2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Safe Merging of Autonomous Vehicles under Temporal Logic Task

Chenguang Zhao, Huan Yu*

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10886277

***Abstract*— Autonomous vehicles are increasingly recognized for their potential to enhance traffic stability and safety through longitudinal control. However, a significant challenge remains in controlling AVs at merging ramps due to increased risks of congestion and accidents. This paper tackles merging control of an AV from an accelerated lane of finite length to mainline traffic of Human-Driven Vehicles (HV). Compared with safe car-following control, the merging control needs to satisfy multiple constraints due to complexity of the task. A novel dynamical model for merging and car-following behaviors is developed for mixed merging traffic. The complex temporal behaviors of the merging AV and car-following HVs are described with signal temporal logic. We integrate safety, task-specific, and physical constraints into a single control barrier function, which provides input constraints to synthesize a safety-critical controller via quadratic programming. Simulations on real traffic merging trajectories validate that the designed controller enhances traffic stability and efficiency with the merging task completed safely within the temporal and spatial constraints.**

I. Introduction

Merging ramps are one main source of traffic congestion and accidents. The recent advancements in autonomous vehicles (AVs) have encouraged effective traffic on highways, where longitudinal controllers are designed to smooth traffic and reduce congestion. However, the merging controller design is still under investigation. In this paper, we consider the merging control design. Specifically, we focus on the most basic scenario with a triplet of vehicles, as shown in Fig. 1, where a merging AV (MV) adjusts its speed and position to merge into the main lane when there is a safe gap between both the leading human-driven vehicle (LV) and following human-driven vehicle (FV).

Compared with longitudinal control where the AV is usually assumed to have enough time and space to adapt to other vehicles, the merging control is time-critical and must be completed within a time constraint given the finite length of the acceleration lane as shown in Fig. 1. Given these constraints, most merging controllers are constructed by solving an optimization problem, where the length of the acceleration lane is one constraint. The travel time is included in the objective function or constraints [8], [9], [15], [17]. Since the optimization problem is constructed based on the end state, i.e., the AV merges into the main lane, it is implemented in a model predictive control (MPC) approach [21].

Chenguang Zhao and Huan Yu are with the Hong Kong University of Science and Technology (Guangzhou), Thrust of Intelligent Transportation, Guangzhou, Guangdong, China. Huan Yu is also affiliated with the Hong Kong University of Science and Technology, Department of Civil and Environmental Engineering, Hong Kong SAR, China.
* corresponding author (huanyu@ust.hk).

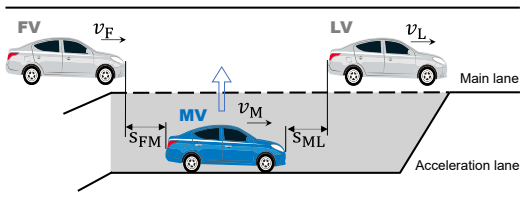

Fig. 1. The merging vehicle (MV) adjusts its speed and gap on the acceleration lane and merges into the main lane when is has a safe gap with both the leading vehicle (LV) and the following vehicle (FV). The merging must be completed before the end of the acceleration lane.

The MPC-based controllers are solved recursively in each time step, which causes a high computation load. Besides, when solving the MPC, prediction of future states until the end of the merging is required. In the merging scenario, the speed of the leading vehicle is an external disturbance that evolves independently from the control input and thus cannot be accurately predicted. Inaccurate prediction will cause conservative and inefficient control strategies.

Besides MPC, merging controllers have also been designed by other methods. In [16], a probabilistic model is proposed where the merging vehicle chooses its trajectory from a set of candidates. For data-driven controllers, a main concern is the lack of a rigorous theoretical guarantee on constraints, such as safety during merging. In [5], a controller is designed with constraints constructed by control barrier function (CBF), which is a tool to guarantee forward invariance of a set, i.e., if the system is within the set initially then it always remains within the set [1]. Compared with MPC which predicts the future control process at each time step, CBF provides constraints based only on the current state and thus has a lower computation load [19], [20]. However, CBF fails to satisfy the time constraint on merging [5].

In this paper, we design a merging controller by temporal logic, which is a formal method developed to express real-time constraints and impose them on time-critical systems [3]. Examples of temporal logic include linear temporal logic, metric temporal logic, and signal temporal logic (STL) [13]. Among these formal languages, we use STL for the merging control since it is evaluated over continuous-time signals and can be equipped with a quantitative robustness metric. A controller that satisfies STL is usually derived by first translating STL formulas into mixed-integer constraints and then solving a mixed-integer optimization problem. Recently, time-varying CBF has been used to encode STL [10], and the control input is solved by quadratic programming. STL has been applied to control AV for multiple goals, such as in obstacle avoidance [6], and safety specification [2]. This

paper presents the first merging controller design by STL.

We identify and express three types of merging constraints by STL syntax: task constraint that the merging should be completed within a maximum duration and the acceleration lane, safety constraint that the MV maintains a safe gap with both LV and FV, physics constraint that the MV speed is positive and smaller than a maximum speed. Constraints on the control input are then correspondingly constructed using CBF. We extract from the Next Generation Simulation (NGSIM) data-set [4] 95 real traffic merging trajectories with a triplet of vehicles as in Fig. 1, and run simulations based on these trajectories. Results show that the designed controller meets all constraints and yields smoother traffic with less speed perturbation. The designed controller also takes a shorter time to finish merging, which increases traffic efficiency. This paper contributes a novel merging controller design method via CBF-STL. Compared with MPC, the CBF-STL has a lower computation load and does not require predictions in the future. Compared with pure CBF, our controller satisfies constraints on merging time.

The remaining part of this paper is organized as follows. We formulate the merging process and specify the three constraints in Section II. In Section III, we give the designed merging controller. In Section IV, we run simulations based on trajectories extracted from the NGSIM data-set to validate the designed controller.

II. Merging Modeling

In this section, we first model the merging process, then we specify the constraints for merging.

A. Vehicle motion modeling

We consider a triple of vehicles, one leading LV and following FV on the main road, and one merging AV (MV). We assume that the sequence of these three vehicles is fixed, i.e., the MV merges into between the LV and FV. This can be decided by first-in-first-out policy, or can be given by upper-level controllers that specify sequencing policies as in [14]. We ignore the lateral motion of the MV, and assume that the merging is finished when there is a proper gap for the MV.

We denote the speed of theses three vehicles as $v_{\mathrm{L}}(t) \in \mathbb{R}$, $v_{\mathrm{F}}(t) \in \mathbb{R}$, and $v_{\mathrm{M}}(t) \in \mathbb{R}$ respectively. The merging AV is controlled by a controller $u(t) \in \mathbb{R}$:

$$\dot{v}_{\mathrm{M}}(t) = u(t), \tag{1}$$

We consider $u(t)$ as the sum of a nominal controller $u_0(t) \in \mathbb{R}$ and an correctness term $\Delta u(t) \in \mathbb{R}$ to facilitate the merging process:

$$u(t) = u_0(t) + \Delta u(t). \tag{2}$$

The nominal controller $u_0(t)$ can be chosen as those pre-designed controllers that have stable car-following behavior or minimize energy consumption [7]. We will solve a minimum correctness term $\Delta u(t)$ that ensures safe merging in this paper so that the merging process has little effect on the main lane traffic. The gap between the FV and the MV $s_{\mathrm{FM}}(t) \in \mathbb{R}$ has the dynamics:

$$\dot{s}_{\mathrm{FM}}(t) = v_{\mathrm{M}}(t) - v_{\mathrm{F}}(t). \tag{3}$$

For the gap between the MV and the LV $s_{\mathrm{ML}}(t) \in \mathbb{R}$, we have:

$$\dot{s}_{\mathrm{ML}}(t) = v_{\mathrm{L}}(t) - v_{\mathrm{M}}(t). \tag{4}$$

For the gap between the FV and LV $s_{\mathrm{FL}}(t) \in \mathbb{R}$, we have:

$$\dot{s}_{\mathrm{FL}}(t) = v_{\mathrm{L}}(t) - v_{\mathrm{F}}(t). \tag{5}$$

In longitudinal car-following, a human driver accelerates based on the state of itself and its leading vehicle, such as the intelligent driver model, and optimal velocity model. In the merging process, the FV should take both the LV and MV as its leaders. To describe such driving behaviors, We introduce an enhanced car-following model to describe FV's motion. The FV's acceleration is:

$$\dot{v}_{\mathrm{F}} = f(v_{\mathrm{F}}, v_{\mathrm{L}}, s_{\mathrm{FL}}, v_{\mathrm{M}}, s_{\mathrm{FM}}). \tag{6}$$

In this paper, we use merging trajectories extracted from the NGSIM data-set to calibrate this model as in Section IV.

B. Constraints design

There are two possible merging policies: prespecified-position merging policy where the merging happens at a fixed merging point and flexible-position merging policy where the AV merges into the main road at a flexible position [12]. We consider the second policy, since in mixed traffic, uncertainties of human drivers may cause limited optimality of the first policy.

We consider three types of constraints:

- Safety constraint. When the MV merges onto the main road, it should keep a safe gap with both the FV and LV. We apply the time to collision (TTC) safe spacing policy to constrain the gap when merging. For a vehicle with speed v and its leader with speed v_l, TTC requires that the two-vehicle should keep a gap

  $$s \geq \tau(v - v_l) + s_{\mathrm{st}}, \tag{7}$$

  with $s_{\mathrm{st}} > 0$ being the minimum safe gap and $\tau > 0$ being the time to risk if the two vehicles remain at the same speed. The MV adjusts its speed and position on the acceleration lane, and merges to the main road once it has a safe gap with both the LV and FV. The merging should be completed within a given time threshold $T > 0$. The safety constraint for the MV is designed as: there exists a $0 \leq t_e \leq T$ such that both the two inequalities hold:

  $$s_{\mathrm{ML}}(t_e) \geq \tau(v_{\mathrm{M}}(t_e) - v_{\mathrm{L}}(t_e)) + s_{\mathrm{st}}, \tag{8}$$
  $$s_{\mathrm{FM}}(t_e) \geq \tau(v_{\mathrm{F}}(t_e) - v_{\mathrm{M}}(t_e)) + s_{\mathrm{st}}. \tag{9}$$

- Task constraint. The MV should merge into the main road before the end of the acceleration lane. We use $p_{\mathrm{M}}(t)$ to represent the location of the AV with $\dot{p}_{\mathrm{M}}(t) = v_{\mathrm{M}}(t)$. We take the beginning of the acceleration lane as

$p_{\mathrm{M}} = 0$, and we assume that $p_{\mathrm{M}}(0) = 0$, i.e., the MV is at the beginning of the acceleration lane initially. The merging task is regarded as finished at t_e if both (8) and (9) hold. The task constraint is given as:

$$p_{\mathrm{M}}(t_e) \leq L. \tag{10}$$

- Physics constraint. Considering traffic regulation policy, the MV cannot drive backward on the acceleration, i.e., for $0 \leq t \leq T$, we have:

$$v_{\mathrm{M}}(t) \geq 0. \tag{11}$$

We also assume that there is a known maximum speed $v_{\max}$, and the MV cannot exceed this upper bound, i.e., for $0 \leq t \leq T$, we have:

$$v_{\mathrm{M}}(t) \leq v_{\max}. \tag{12}$$

III. Merging Controller Design

In this section, we present the designed merging controller. We first give preliminaries on STL and CBF, then design constraints on the control input to satisfy merging constraints.

A. STL language

STL is a predict logic based on predicts μ that are obtained after evaluation of a prediction function h [13]:

$$\mu = \begin{cases} \text{True}, & \text{if } h(x) \geq 0, \\ \text{False}, & \text{if } h(x) < 0, \end{cases} \tag{13}$$

The STL syntax syntax defines an STL formula ϕ:

$$\phi ::= \text{True} \,|\mu\, |\neg\phi\, |\phi_1 \wedge \phi_2\, \big|\phi_1 U_{[t_1,t_2]} \phi_2\ , \tag{14}$$

where ϕ_1 and ϕ_2 are STL formulas, and $0 \leq t_1 < t_2$. The satisfaction relation $(x,t) \models \phi$ denotes if the signal $x : \mathbb{R}_{\geq 0} \to \mathbb{R}$ satisfies formula ϕ at time t.

For a signal $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, the STL semantics are recursively given by:

$$\begin{aligned}
&(x,t) \models \mu && \Leftrightarrow h(x(t)) \geq 0 \\
&(x,t) \models \neg\phi && \Leftrightarrow \neg\left((x,t) \models \phi\right) \\
&(x,t) \models \phi_1 \wedge \phi_2 && \Leftrightarrow (x,t) \models \phi_1 \wedge (x,t) \models \phi_2 \\
&(x,t) = \phi_1 U_{[t_1,t_2]} \phi_2 && \Leftrightarrow \exists t_3 \in [t+t_1, t+t_2], (x,t_3) \models \phi_2 \\
& && \quad \wedge \forall t_4 \in [t,t_3], (x,t_4) \models \phi_1 \\
&(x,t) \models F_{[t_1,t_2]} \phi && \Leftrightarrow \exists t_3 \in [t+t_1, t+t_2], (x,t_3) \models \phi \\
&(x,t) \models G_{[t_1,t_2]} \phi && \Leftrightarrow \forall t_3 \in [t+t_1, t+t_2], (x,t_3) \models \phi
\end{aligned}$$

B. Time varying CBF

Consider a general control affine system with state $x \in \mathbb{R}^{n_1}$ and control input $u \in \mathbb{R}^{n_2}$:

$$\dot{x} = f(x) + g(x)u, \tag{15}$$

with f and g being locally Lipschitz continuous. We assume that it has a unique solution on the time interval $[t_0, t_1]$ with $t_1 > t_0 \geq 0$ given an initial condition $x(t_0)$.

Definition 1 (Time-varying control barrier function)**.** *A differentiable function $b(x,t)$ defined on $\mathcal{D} \times [t_0, t_1]$ with $\mathcal{D} \subset \mathbb{R}^{n_1}$ is a time-varying CBF if there exists a locally Lipschitz continuous class $\mathcal{K}$ function α so that, $\forall (x,t) \in \mathcal{D} \times [t_0, t_1]$,*

$$\sup_u \frac{\partial b(x,t)}{\partial x}(f(x) + g(x)u) + \frac{\partial b(x,t)}{\partial t} \geq -\alpha(b(x,t)). \tag{16}$$

Theorem 1 (Forward invariance by CBF [18])**.** *If $b(x,t)$ is a time-varying CBF, then any locally Lipschitz continuous controller u satisfying*

$$\frac{\partial b(x,t)}{\partial x}(f(x) + g(x)u) + \frac{\partial b(x,t)}{\partial t} \geq -\alpha(b(x,t)), \tag{17}$$

renders the set $\mathcal{C}(t) = \{x | b(x,t) \geq 0\}$ forward invariant, i.e., if $x(t_0) \in \mathcal{C}(t_0)$, then $x(t) \in \mathcal{C}(t)$ for all $t \in [t_0, t_1]$.

Example 1 (CBF for *eventually* STL task F)**.** *Given a predict μ with the prediction function $h(x)$ with $\max_x h(x) > 0$, if we have an STL $(x,0) \models F_{[0,t_1]}\mu$, construct a time-varying CBF as*

$$b(x,t) = h(x) - \gamma(t), \tag{18}$$

where the continuous function $\gamma(t)$ defined on $t \in [0, t_1]$ satisfies $\gamma(0) < h(x(0))$ and $\gamma(t_1) > 0$, then if the controller satisfies

$$L_f h(x) + L_g h(x) u - \dot{\gamma}(t) \geq -\alpha(b(x,t)), \tag{19}$$

where $L_f h(x) = \nabla h(x) \cdot f(x)$, $L_g h(x) = \nabla h(x) \cdot g(x)$ are Lie derivatives, we have $(x,0) \models F_{[0,t_1]}\mu$.

There are multiple choices for the function $\gamma(t)$. One simple choice is a linear function [11]:

$$\gamma(t) = \frac{\gamma_\infty - \gamma_0}{t_1} t + \gamma_0, \tag{20}$$

with $\gamma_0 < h(x(0))$ and $0 < \gamma_\infty < \max_x h(x)$. By setting a positive $\gamma_\infty > 0$, we have $h(x(t_1)) > \gamma_\infty > 0$, which means the STL is met with a robustness γ_∞.

Example 2 (CBF for *always* STL task G)**.** *Given a predict μ with the prediction function $h(x)$ with $\max_x h(x) > 0$, if we have an STL $(x,0) \models G_{[0,t_1]}\mu$[1], construct a time-varying CBF as*

$$b(x,t) = h(x) - \gamma(t), \tag{21}$$

where the continuous function $\gamma(t)$ defined on $t \in [0, t_1]$ satisfies $\gamma(0) < h(x(0))$ and $\gamma(t) > 0$ for all $t \in [0, t_1]$, then if the controller satisfies

$$L_f h(x) + L_g h(x) u - \dot{\gamma}(t) \geq -\alpha(b(x,t)), \tag{22}$$

we have $(x,0) \models G_{[0,t_1]}\mu$.

For the *always* STL syntax G, pure CBF with $\gamma(t) \equiv 0$ also works. For the *eventually* STL syntax F, if $h(x(0)) < 0$, pure CBF with $\gamma(t) \equiv 0$ fails to meet the control goal, and a function $\gamma(t)$ is required. This is because CBF only guarantees forward invariance of a set when the initial state is within the set. If $h(x(0)) < 0$, CBF no longer

[1]We assume that $h(x(0)) \geq 0$, otherwise the STL semantic cannot be achieved.

ensures $h(x(t))$ is always non-negative. But by choosing $\gamma(0) < h(x(0))$, we have $b(x,0) > 0$, and time-varying CBF guarantees $b(x,t) \geq 0$ holds for any $t \in [0, t_1]$.

In the merging control, we specify three constraints, task constraint, safety constraint, and physics constraint. The task constraint specifies constraints only on the final state when the merging is finished, and we use *eventually* STL task. For safety and physics constraints, since they involve constraints during the whole merging process, the *always* STL task is adopted.

C. Control constraints design

We design constraints on the MV control input u corresponding to the three types of merging constraints.

Safety constraint: For the safe gap between MV and LV (8), we construct a predict function as:

$$h_{\mathrm{M}} = s_{\mathrm{ML}} - \tau_{\mathrm{M}}(v_{\mathrm{M}} - v_{\mathrm{L}}) - s_{\mathrm{st}}. \tag{23}$$

We choose $\gamma_{\mathrm{M}}(t)$ as a linear function as in Example 1 with $\gamma_{\mathrm{M}}(0) < h_{\mathrm{M}}(0)$ and $\gamma_{\mathrm{M}}(T) > 0$, and construct:

$$b_{\mathrm{M}}(x,t) = h_{\mathrm{M}}(x) - \gamma_{\mathrm{M}}(t). \tag{24}$$

Similarly, for the safe gap between FV and MV (9), we construct a predict function as:

$$h_{\mathrm{F}} = s_{\mathrm{FM}} - \tau_{\mathrm{F}}(v_{\mathrm{F}} - v_{\mathrm{M}}) - s_{\mathrm{st}}, \tag{25}$$

and choose a proper function $\gamma_{\mathrm{F}}(t)$ to get the corresponding time-varying CBF candidate as:

$$b_{\mathrm{F}}(x,t) = h_{\mathrm{F}}(x) - \gamma_{\mathrm{F}}(t). \tag{26}$$

Task constraint: Given the task constraint (10), a corresponding predict function is:

$$h_l = L - p_{\mathrm{M}}. \tag{27}$$

The corresponding time-varying CBF is

$$b_l(x,t) = h_l(x) - \gamma_l(t). \tag{28}$$

The task constraint requires that the MV is before the end of the acceleration lane when there is a safe gap. While setting $\gamma_l(T) \geq 0$ as in Example 1 ensures the task constraint is met, it may cause conservative strategies since the MV has already merged into the main lane at T. Enforcing $\gamma_l(T) > 0$ actually requires the MV must travel less than L before T even if it has merged into the main lane. We design a less conservative γ_l as follows: Since $\gamma_{\mathrm{M}}(t)$ is a linear function with $\gamma_{\mathrm{M}}(T) > 0$, so there exists a $t_{e\mathrm{M}} \in [0, T)$ such that $\gamma_{\mathrm{M}}(t) > 0$ holds for all $t \in [t_{e\mathrm{M}}, T]$. Similarly, there exists a $t_{e\mathrm{F}} \in [0, T)$ for γ_{F}. If both (24) and (26) are satisfied, we have $\hat{t}_e = \max\{t_{e\mathrm{M}}, t_{e\mathrm{F}}\}$ that ensures there is a safe gap at $\hat{t}_e$. So instead of requiring $\gamma_l(T) > 0$, the function $\gamma_l(t)$ is chosen with $\gamma_l(0) < h_l(0)$ and $\gamma_l(\hat{t}_e) > 0$.

We note that $\dot{h}_l = -v_{\mathrm{M}}$, i.e., h_l has a relative degree two with respect to the control input u. So $b_l(x,t)$ does not provide any explicit constraint on the control input. Since

$$\dot{b}_l(x,t) \geq -\alpha_l(b_l(x,t)) \implies v_{\mathrm{M}}(t) \leq \alpha_l(b_l(x,t)) - \dot{\gamma}_l(t), \tag{29}$$

with α_l being a class $\mathcal{K}$ function, the task constant is satisfied if (29) is met. We further construct a time-varying CBF as:

$$b_{\bar{l}}(x,t) = h_{\bar{l}}(x) - \gamma_{\bar{l}}(t), \tag{30}$$

where $h_{\bar{l}} = \alpha_l(b_l) - v_{\mathrm{M}}$, $\gamma_{\bar{l}} = -\dot{\gamma}_l + \Delta\gamma_l$ with $\Delta\gamma_l$ being a continuous time-varying function that has $\Delta\gamma_l(0) + \dot{\gamma}_l(0) < h_{\bar{l}}(x(0))$ and $\Delta\gamma_l(\hat{t}_e) + \dot{\gamma}_l(\hat{t}_e) > 0$.

Physics constraint: Given the minimum speed (11) and maximum speed (12), two predict functions are constructed:

$$h_{\underline{v}} = v_{\mathrm{M}}, \tag{31}$$

$$h_{\bar{v}} = v_{\max} - v_{\mathrm{M}}. \tag{32}$$

By choosing proper $\gamma_{\underline{v}}(t)$ and $\gamma_{\bar{v}}(t)$), two time-varying CBFs are respectively constructed as:

$$b_{\underline{v}}(x,t) = h_{\underline{v}}(x) - \gamma_{\underline{v}}(t), \tag{33}$$

$$b_{\bar{v}}(x,t) = h_{\bar{v}}(x) - \gamma_{\bar{v}}(t). \tag{34}$$

D. QP formulation

The merging constraints give five CBF candidates: (24), (26), (30), (33), and (34). We synthesize one overall time-varying CBF based on these five CBFs as:

$$b(x,t) = -\frac{\ln\left(e^{-\eta b_{\mathrm{M}}} + e^{-\eta b_{\mathrm{F}}} + e^{-\eta b_{\bar{l}}} + e^{-\eta b_{\underline{v}}} + e^{-\eta b_{\bar{v}}}\right)}{\eta}, \tag{35}$$

where $\eta > 0$ is a parameter that satisfies $b(x,0) \geq 0$. Since

$$b(x,t) \leq \min_i b_i(x,t), \tag{36}$$

for any $i = M, F, \bar{l}, \underline{v}, \bar{v}$, so if the controller satisfies

$$\frac{\partial b(x,t)}{\partial x}\dot{x} + \frac{\partial b(x,t)}{\partial t} \geq -\alpha(b(x,t)), \tag{37}$$

then $b(x,t) \geq 0$, which implies all five constraints are satisfied. The time derivative of the overall time-varying CBF $b(x,t)$ is

$$\begin{aligned} &\frac{\partial b(x,t)}{\partial x}\dot{x} + \frac{\partial b(x,t)}{\partial t} \\ =&\sum_i \frac{\partial b_i(x,t)}{\partial x}\dot{x} + \frac{\partial b_i(x,t)}{\partial t} \\ =&\sum_i e^{-\eta(b_i - b)}\left(L_f h_i(x) + L_g h_i(x) u - \dot{\gamma}_i(t)\right). \end{aligned} \tag{38}$$

We solve a merging controller via the quadratic programming:

$$\begin{aligned} \Delta u = \underset{u \in \mathbb{R}}{\operatorname{argmin}}\ & |u|^2, \\ \text{s.t. } & \frac{\partial b(x,t)}{\partial x}\dot{x} + \frac{\partial b(x,t)}{\partial t} \geq -\alpha(b(x,t)), \end{aligned} \tag{39}$$

with α being a class $\mathcal{K}$ function.

TABLE I

EVALUATION METRIC

	HV	AV	AV improvement
FV acceleration $\bar{a}_{\rm F}$ (m/s^2)	0.67	0.60	9.45%
MV acceleration $\bar{a}_{\rm M}$ (m/s^2)	0.64	0.46	27.97%
Merging time T_m (s)	5.00	2.79	44.10%

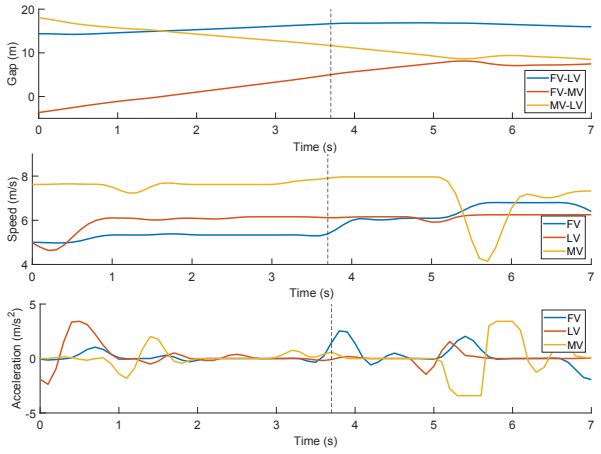

Fig. 2. First case trajectories extracted from the NGSIM data-set.

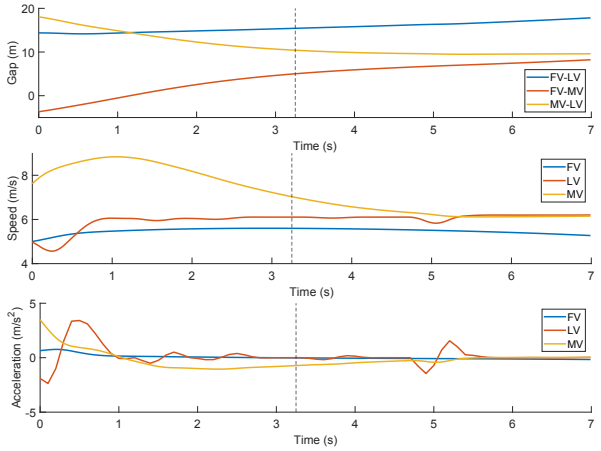

Fig. 3. First case trajectories by the designed merging controller (39).

IV. SIMULATION VALIDATION

We use the trajectory data in NGSIM collected from 05:00 to 05:15 pm on the I-80 highway. The trajectories record the longitudinal and lateral position, speed, and acceleration of each vehicle. The data-set has also given the lane information, with Lane 7 as the on-ramp merge lane, and Lane 6 as the farthest right lane which vehicles merge into.

We extract 95 merging trajectories and run simulations based on these real trajectories. We set the initial speed of the three vehicles, i.e., $v_{\rm M}(0)$, $v_{\rm L}(0)$, $v_{\rm F}(0)$, and the initial gap, i.e., $s_{\rm ML}(0)$, $s_{\rm FL}(0)$, $s_{\rm FL}(0)$, the same as in the real trajectories. We set the leading vehicle's dynamics $\dot{v}_{\rm L}$ the same as in the data-set. For the extended car-following model (6), we use the real trajectories to calibrate a linear model. More accurate model calibration is left as one future work. We set the safe time to collision (8) as $\tau = 1$ s and the minimum safe gap as $s_{\rm st} = 5$ m. We choose the parameter $\eta = 1$, which satisfies $b(x, 0) > 0$. For the class $\mathcal{K}$ function function, we choose $\alpha(x) = 10x$. For the nominal controller of the merging vehicle, we set it as the car-following type controller [7]:

$$u_0 = a(V(s_{\rm ML}) - v_{\rm M}) + b(v_{\rm L} - v_{\rm M}), \tag{40}$$

where $a > 0$ and $b > 0$ represent how the vehicle accelerates to match its speed $v_{\rm M}$ with the spacing-dependent desired speed $V(s_{\rm ML})$ and its leader speed $v_{\rm L}$ respectively. We use the desired speed as

$$V(s) = \begin{cases} 0, & s \le s_{\rm st}, \\ v_{\max}\frac{s - s_{\rm st}}{s_{\rm go} - s_{\rm st}}, & s_{\rm st} < s < s_{\rm go}, \\ v_{\max}, & s \ge s_{\rm go}, \end{cases} \tag{41}$$

We set the parameters as $a = 0.6$, $b = 0.9$, $s_{\rm go} = 35$ m, and $v_{\max} = 40$ m/s.

We compare the designed controller with HVs in Table I. For each simulation, we calculate the average of absolute value of acceleration during the merging process for the MV and FV as $\bar{a}_{\rm M}$ and $\bar{a}_{\rm F}$ respectively. Table I gives the average $\bar{a}_{\rm M}$ and $\bar{a}_{\rm F}$ over 95 scenarios. By the designed controller, both the MV and FV have a lower $\bar{a}_{\rm M}$ and $\bar{a}_{\rm F}$, which means the speed perturbation is smaller and thus the traffic stability is enhanced. We also calculate the average time to finish merging T_m for the 95 scenarios in Table I. The designed controller takes less time to merge.

We further extract two representative merging scenarios to analyze how the controller works. Fig. 2 and Fig. 4 give HV trajectories from the data-set for the two scenarios. The dotted vertical line gives the time instant when the MV merges into the main lane. In these two scenarios, the HV takes different diving strategies to merge. In the first case as in Fig. 2, the merging vehicle merges at a higher speed than the LV and then decelerates to match the main lane traffic after merging. In the second case as shown in Fig. 4, the merging vehicle tries to match LV's speed on the acceleration lane and then merges.

We give in Fig. 3 the simulated trajectories of the three vehicles in the first merging case. The merging is finished around 3 sec as the dotted vertical line shows. Compared the trajectories by the designed merging controller (39) and by human drivers from the data-set, we see that the controlled merging is smoother with a smaller speed perturbation. For example, in Fig. 2, the MV merges to the main lane at a higher speed than the LV and MV, and then sharply decelerates and accelerates during 5 to 6 sec to adapt to the main lane traffic with a speed perturbation form 8 m/s to 4 m/s and then to 7 m/s. While in Fig. 3, the merging vehicle adjusts its speed before merging, and has a much smaller acceleration (less than $\pm 1 {\rm m/s}^2$) after it merges into the main lane.

In Fig. 5, we plot trajectories by the designed controller in the second merging case. Compared with human drivers in Fig. 4, we see that the controlled MV presents a smoother acceleration perturbation, which also increases the stability of the following vehicle. As shown in Fig. 4, the merging

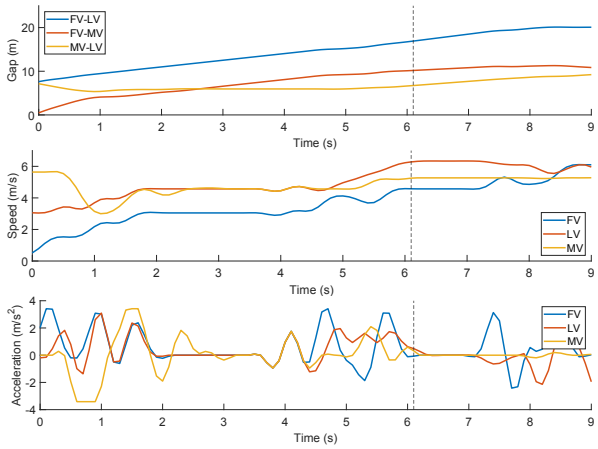

Fig. 4. Second case trajectories extracted from the NGSIM data-set.

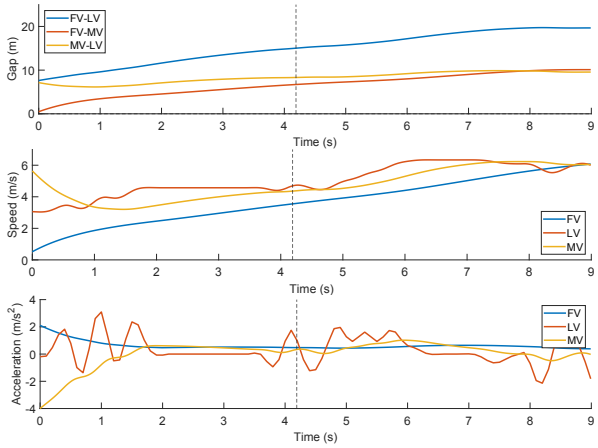

Fig. 5. Second case trajectories by the designed merging controller (39).

vehicle has a larger initial speed than the leading vehicle. The merging vehicle first decelerates to a lower speed and then keeps almost the same speed as the leading vehicle during 2.5 to 4.5 sec, and finally decelerates to merge at 6 sec. This leads to a longer merging time since the gap between the MV and LV is almost constant during 2.5 to 4.5 sec, and the MV has to decelerate to create a safe merging gap. The controlled MV, as shown in Fig. 5, first decelerates and then drives at a lower speed than the LV. As a result, the merging takes a shorter time of around 4 sec.

V. Conclusion

In this paper, we consider safe merging controller design for an autonomous vehicle. Temporal logic STL syntax is used to express three types of merging constraints, and time-varying CBFs encode the STL syntax as control constraints. A safe merging controller is solved by a QP to minimize the deviation from a nominal human merging control. Simulations based on NGSIM traffic trajectories validate that the designed controller has smoother merging. Future extension of this work includes joint optimization of sequencing policy, safety requirements and stabilizing control design for generic mixed-autonomy scenarios.

References

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[2] N. Arechiga, "Specifying safety of autonomous vehicles in signal temporal logic," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 58–63.

[3] C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 115–140, 2019.

[4] FHWA, "Next Generation Simulation (NGSIM)," 2007. [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[5] V. Hamdipoor, N. Meskin, and C. G. Cassandras, "Safe merging control in mixed vehicular traffic," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 4386–4392.

[6] Z. Huang, W. Lan, and X. Yu, "A formal control framework of autonomous vehicle for signal temporal logic tasks and obstacle avoidance," *IEEE Transactions on Intelligent Vehicles*, 2023.

[7] I. G. Jin, S. S. Avedisov, C. R. He, W. B. Qin, M. Sadeghpour, and G. Orosz, "Experimental validation of connected automated vehicle design among human-driven vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 335–352, 2018.

[8] M. Karimi, C. Roncoli, C. Alecsandru, and M. Papageorgiou, "Cooperative merging control via trajectory optimization in mixed vehicular traffic," *Transportation Research Part C: Emerging Technologies*, vol. 116, p. 102663, 2020.

[9] V.-A. Le, H. M. Wang, G. Orosz, and A. A. Malikopoulos, "Coordination for connected automated vehicles at merging roadways in mixed traffic environment," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 4150–4155.

[10] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 1, pp. 96–101, 2019.

[11] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[12] H. Liu, W. Zhuang, G. Yin, Z. Li, and D. Cao, "Safety-critical and flexible cooperative on-ramp merging control of connected and automated vehicles in mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 2920–2934, 2023.

[13] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *International symposium on formal techniques in real-time and fault-tolerant systems*. Springer, 2004, pp. 152–166.

[14] E. Sabouni and C. G. Cassandras, "Optimal merging control of an autonomous vehicle in mixed traffic: An optimal index policy," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2353–2358, 2023.

[15] N. Venkatesh, V.-A. Le, A. Dave, and A. A. Malikopoulos, "Connected and automated vehicles in mixed-traffic: Learning human driver behavior for effective on-ramp merging," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 92–97.

[16] E. Ward, N. Evestedt, D. Axehill, and J. Folkesson, "Probabilistic model for interaction aware planning in merge scenarios," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 133–146, 2017.

[17] W. Xiao, C. G. Cassandras, and C. A. Belta, "Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles," *Automatica*, vol. 129, p. 109592, 2021.

[18] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018.

[19] C. Zhao and H. Yu, "Robust safety for mixed-autonomy traffic with delays and disturbances," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[20] C. Zhao, H. Yu, and T. G. Molnar, "Safety-critical traffic control by connected automated vehicles," *Transportation research part C: emerging technologies*, vol. 154, p. 104230, 2023.

[21] J. Zhu, S. Easa, and K. Gao, "Merging control strategies of connected and autonomous vehicles at freeway on-ramps: A comprehensive review," *Journal of intelligent and connected vehicles*, vol. 5, no. 2, pp. 99–111, 2022.